[redacted]

25X1

Attachment:
OSI-RP/73-5 July 1973

SECRET

Soviet Computer Software

Secret
OSI-RP/73-5
July 1973

Copy № 28

25X1

[redacted] 25X1

RESEARCH PAPER

SOVIET COMPUTER SOFTWARE

Project Officer

[redacted] 25X1

OSI-RP/73-5
July 1973

CENTRAL INTELLIGENCE AGENCY
DIRECTORATE OF SCIENCE AND TECHNOLOGY
OFFICE OF SCIENTIFIC INTELLIGENCE

[redacted] 25X1

[redacted] 25X1

PREFACE

This research paper discusses Soviet efforts to develop computer software, especially in the areas of computer languages, compilers, translators, and operating systems. (Detailed information in these areas is available upon request in three additional appendices.) The report also analyzes Soviet capabilities to use computer languages in specific types of applications.

[redacted] 25X1

[redacted] 25X1

[redacted]

TECHNICAL FOREWORD

The advent of the computer called for many innovations, not the least of which was the need for man to learn to communicate with the machine. This required that man be conversant with machine language--a language in which number sequences represent data and signify manipulations to be performed.

A programming language allows the computer user to communicate what he wants the machine to do and how he wants the results displayed. A compiler or translator for a programming language then converts these semiconversational statements into the more complex language of the machine for execution. An operating system allocates the various resources of the computer to allow for their most efficient use and for accurate transferral of results to the appropriate output devices.

Certain terms used throughout the report are defined* below.

Software is a set of computer programs, procedures, and possibly associated documentation concerned with the operation of a data processing system (e.g., compilers, library routines, manuals, flow charts).

Hardware means physical equipment (e.g., mechanical, magnetic, electrical, or electronic devices)--as opposed to the computer program or method of use.

Programming language is a set of representations, formats, and rules used to communicate the procedure for problem solution. Specifically, it is a language used to prepare computer programs. There are several types of programming languages:

Machine language consists of a series of numbers which are accepted by a computer and cause a specific operation to take place. These numbers, when expressed in binary form, have a pattern that represents the pattern of the electronic signals actually used in the computer

*See American National Standard: Vocabulary for Information Processing, ANSI X3.12-1970, published by the American National Standards Institute, Inc. New York, and CDP Review Manual: A Data Processing Handbook, edited by Roger A. MacGowan and Reid Henderson, Auerbach Publishers, Inc., USA, 1972.

[redacted]

circuits. Machine language programs are unique to a specific computer model and require specialized knowledge on the part of the user.

Symbolic language (also referred to as an assembly language) utilizes alphabetical, numerical, and special characters to identify and to name data and instruction locations (mnemonic addresses). Symbolic languages are machine dependent and must be converted to machine languages before they can be used on a computer. This is accomplished by an assembler. Although programs written in symbolic language are unique to a given model of computer, they bear a closer resemblance to the natural language of the user than do machine languages.

Higher level language (also referred to as a compiler language) is designed for the convenient expression of procedures used in the solution of a wide class of problems. Even though higher level languages are artificial, their syntax and grammar are similar to those of a natural language. In addition, such languages are said to be machine independent since they can be translated into many different machine languages. As with assembly languages, programs written in higher level languages (source programs) must be translated into machine languages (object programs) before they can be executed on a computer. A single higher level language statement may correspond to a sequence of machine language statements. Higher level languages are important because they offer a convenient means of stating a problem and solution procedure without requiring specialized knowledge of the computer to be used. In most cases higher level languages have been developed to incorporate features that make them particularly suited to a particular class of problems.

In order to automate the conversion of programs written in one type of language into another, there is a need for certain types of programs that are independent of any specific problem-solving application. The four most important types of conversion programs, usually considered to be a part of a computer system's software, are:

Assembler, a computer program that prepares a machine language program from a symbolic language program by substituting absolute operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

Compiler, a computer program that prepares a machine language program from a computer program

[redacted]

written in another programming language by making use of the overall logic structure of the latter, or generating more than one machine instruction for each symbolic statement, or both, and by performing the function of an assembler.

Interpreter, a computer program that translates and executes each source language statement before translating and executing the next one.

Translator, a computer program that transforms statements from one language to another without significantly changing the meaning.

Because computers usually consist of a number of functional devices, there is a need for certain specialized types of programs that manage the system and make it perform in the desired fashion regardless of the programs being run.

An operating system is that computer software which controls the execution of computer programs. An operating system may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services. Obviously, the more complex the computer system, the greater the need for an operating system and the more difficult the job of developing a suitable operating system.

[redacted]

[redacted]

25X1

CONTENTS

TABLES

25X1

[redacted]

[redacted]

FIGURE

[redacted]

[redacted]

SOVIET COMPUTER SOFTWARE

PROBLEM

To assess Soviet capabilities to provide for the software needs of the USSR, particularly for military and other advanced computer systems.

CONCLUSIONS

1. Soviet capabilities to provide for the software needs of complex military systems are only marginal. Specifically, the Soviets have adequate capabilities to provide software for guidance systems, but they have only limited software capabilities for supporting tactics or logistics systems. They now are incapable of supporting the software needs of a complex command and control system composed of networks of computers operating online, particularly in a time-sharing mode. Software for Soviet military systems is probably no further advanced than that of the civilian sector, with most military computer programs written in machine or assembly language. Military interests have been identified with work on multipurpose languages, computer-aided-design languages, and time-sharing operating systems, but the extent of military application is unknown.

2. Soviet higher level programming language and related compiler developments are characterized by independent, and at times redundant, efforts. They have been geared largely toward scientific problem-solving applications, with the ALGOL-60 language and its derivatives receiving most attention. Higher level language compilers have been developed for most models of domestic computers, but machine language programming predominates. Soviet capabilities in this area have been hindered by ineffective means for program sharing, and software documentation and maintenance are poor.

3. Soviet development of standard types of computer systems software has been impeded by shortcomings in coordination and by the limited storage and input/output capabilities of domestic computers. Parochialism and differences in computer system configuration among facilities have resulted in independent and localized system software development. Soviet interest in time-sharing operating systems is high but actual accomplishments have not kept pace with the demonstrated theoretical understanding, and only experimental systems have been developed. Activities in this area parallel those of the US in the early 1960s. System software developments continue to suffer from an insufficient base of computer programmers experienced in the use of complex computer systems.

SUMMARY

Development of Soviet computer software has been characterized by independent, uncoordinated efforts. These efforts have been directed largely toward satisfying the needs of local institutions and their individual computers. Well before 1960 the Soviets demonstrated a theoretical understanding comparable with that of their Western contemporaries on the problems associated with computer software development, but the software that the Soviets have produced has not kept pace with the needs of complex systems. For example, back-up documentation has been inadequate for easy location and correction of errors. Maintenance software has not been developed and thoroughly documented concurrently with corresponding hardware. Space missions have failed because of software errors, probably caused by inadequate program check-out procedures. In order to close the gap, Soviet designers will have to emphasize the development of general purpose time-sharing and multiprocessing operating systems that can easily support compilers for high level languages. The Soviets will also have to provide for more effective user groups to help solve software problems. They will also have to accelerate efforts to build a sizable base of programmers who are experienced in the use of higher level languages and in the operation of complex computer systems.

The total extent of military involvement in Soviet software developments is not known, but the military affiliations of people and institutes involved in these efforts indicate possible military applications work on multipurpose languages, system design languages, and time-sharing operating systems. The Soviets are capable of developing adequate software to support guidance systems, scientific and technical problem-solving, and process control systems. Soviet development of computer-aided design systems will continue to be hampered by the limited capabilities of Soviet computers for input/output and storage. These hardware limitations and the lack of experience with automatic systems for mass data manipulation will impede the development of software for complex command and control, tactical, logistical, and other nonnumeric data processing systems.

The Soviet Union and its East European neighbors are currently developing a series of general purpose computers called RYAD which is similar to the IBM/360 series. The Soviets have not demonstrated an adequate understanding of the problems associated with 360-type software, however, and appear to be repeating mistakes others have made in developing software for similar types of computers. Sufficient emphasis is not being placed on some software areas such as Job Control Language, used to initiate and control job processing and to provide data characteristics and peripheral equipment requirements at the time a program is executed. It is likely, therefore, that the RYAD project will not be the panacea that the Soviets expect it to be and

that lack of experience will cause them great difficulty in developing computer programs for such applications as national economic data processing.

The first significant high level programming language to be implemented by the Soviets was ALGOL-60, which became the nucleus for most later software efforts. Most of the 60-odd higher level languages and the 50 compilers that have been identified are extensions or specialized subsets of ALGOL-60. The remainder are one-of-a-kind languages for which no compilers have been identified. The Soviet decision to use ALGOL-60 was probably based in part on the international effort that went into its development. In addition, ALGOL has a strong mathematical orientation, and most Soviet software developers are mathematicians. The use of ALGOL-60 and ALGOL-based languages probably will continue as the Soviets increase their efforts to develop and implement multipurpose languages, particularly PL/1. Multipurpose languages like PL/1 probably will be increasingly utilized with the RYAD computer series.

Early Soviet language compilers appeared to be straightforward and unsophisticated, geared more toward getting the job done than toward improving the techniques used. Most compilers were designed to accommodate the needs of local installations and the specific computer model being employed. In several cases multiple compilers existed for the same model of computer, indicating a lack of coordination between organizations as well as an absence of effective computer user groups. Most programs were written in machine or assembly language, very few in higher level languages. The literature indicates that most of these practices still prevail and probably will continue for the current models of Soviet computers.

The Soviets have expressed their intention to exploit thoroughly proven Western computer languages such as FORTRAN and COBOL. The mathematically oriented Soviet software designers previously were opposed to the use of COBOL for economic data processing but the Soviets now intend to extend its use in order to benefit from large Western program libraries. Increased utilization of FORTRAN will likewise enable them to benefit from the Western scientific program libraries already in existence.

The majority of operating systems for Soviet computers has been developed by individual institutes. These operating systems are handicapped by the relatively small internal memory capabilities and poor peripheral equipment of the computers involved. Until recently, magnetic disc units were nonexistent and operating systems were based exclusively on the use of magnetic tapes or magnetic drums. Complex operating systems could not be supported and indeed were not needed since programs were written largely in machine or assembly language. Even now the Soviets do not have enough well-trained experienced computer specialists to develop and maintain standardized operating systems.

Operating system expertise in the Soviet Union appears to be concentrated in a few facilities, particularly in the Institute of Applied Mathematics (IAM) at the Academy of Sciences in Moscow. The relatively small IAM group was to have developed a standard operating system for the 1965-vintage BESM-6 computer, but this group has not yet completed its task satisfactorily--probably because of managerial inadequacy and differences in machine configuration at various installations. As a result, several organizations--notably the Institute of Cybernetics in Kiev and the Joint Institute of Nuclear Research at Dubna--have had to provide operating systems for the BESM-6 on the basis of their local needs.

Soviet interest in developing general purpose time-sharing operating systems has increased greatly. Current high level plans to increase utilization of computer systems and to develop a nationwide network of computers and automated management information systems have imposed increased requirements for development of general purpose time-sharing systems. While the Soviets have claimed the time-sharing capability for a number of their operating systems, available material indicates only experimental or very simplistic developments at scientific research and development establishments. Lack of appropriate hardware and peripheral equipment is a major handicap to software development efforts in this area.

DISCUSSION

Background

Even before 1960, such Soviet computer pioneers as A. P. Yershov, A. I. Kitov, L. V. Kantorovich, and A. A. Lyapunov demonstrated strong capabilities in the mathematical and theoretical aspects of computer software. (1-9) In particular, Lyapunov's theory of programming has been a classic in the field. (8) A revised version of the textbook on computer programming by A. I. Kitov and N. A. Krinitskiy is still one of the works most frequently cited. (9)

Prior to 1965 machine language was used almost entirely for programming, and high level language development in the USSR consisted of basic independent research by many individuals. Soviet software designers were so inexperienced that programs were not designed for step-by-step development nor were provisions made for possible program expansion. (10) A. P. Yershov and his associates at the Computing Center of the USSR Academy of Sciences at Novosibirsk devoted 35 man years to development of the ALGOL-dialect ALPHA language and its associated compiler. (11) M. R. Shura-Bura became involved with ALGOL implementation and with general programming theory. V. M. Glushkov authored books and articles on cybernetics and abstract programming theory. (13-20) Other programming systems developed before 1965 apparently were specialized research tools that never became popular. These included the PSK-1 compiling system, which utilized a subroutine library, and various developmental efforts for interpreting systems and primitive assemblers. (21-24)

Since 1965 the Soviets have intensified on-going development programs in the area of computer languages and language compilers. Although good theoretical work has been demonstrated, practical work has been constrained by the limited capabilities of internal and auxiliary stores and by lack of good input/output equipment for both alphabetical and numerical data. (25) Most early efforts were directed toward development of scientific and computational languages at the expense of business and data processing languages and associated compilers. As the numbers of computers have increased and as their use for economic planning and business-oriented as well as scientific applications has expanded, software problems have likewise assumed greater importance.

Languages and Compilers

Soviet computer languages are divided into six major categories:

Numerical scientific
Data processing
String and list processing
Multipurpose
System design
Miscellaneous

Only the more significant features of these languages and compilers are discussed here.

NUMERICAL SCIENTIFIC LANGUAGES

These languages, which are oriented toward computational capabilities were originally designed to solve such engineering problems as matrix inversion and to evaluate mathematical formulas. Twenty-five languages mentioned in the Soviet literature are categorized as numerical scientific languages, five of which--ALCOR, ALGOL, FORTRAN, CERN-FORTRAN, and SUBSET-ALGOL--were developed in Western countries. There are more compilers (approximately 24) in the numerical scientific category than in any other category, which indicates a strong Soviet interest in numerical problem-solving.

ALGOL is the most widely used high level language in the Soviet Union, probably partly due to the international effort that produced it (it is used extensively in Europe) and partly because of its strong mathematical orientation (more appealing to mathematicians than the languages oriented more toward engineering). ALGOL was created to solve a variety of problems in engineering and numerical analysis, but it has also been used for numerical forecasts, processing of observation data, and certain information retrieval applications. (26) [redacted] 25X1

25X1 [redacted]

25X1 [redacted] For instance, the translator of the SUBSET ALGOL-60 language, developed at the Computing Center of the Moscow State University, is part of the BESM-6 operating system there.

Under the leadership of A. P. Yershov at the Academy of Sciences Computing Center in Novosibirsk, the Soviets have been major participants in international meetings held to determine specifications for the updated version of ALGOL, called ALGOL-68. Apparently no ALGOL-68 compilers have been developed for Soviet computers. (28)(29)(38) The increased complexity of the language requires greater storage and input/output capabilities than are generally available on Soviet computers.

The ALPHA language is the second most widely publicized Soviet computer language; it is reportedly in use all over the USSR. (11) A. P. Yershov and his associates developed ALPHA, which is an extension of ALGOL-60, in 1959-64. In order to make better use of their computers, new features--handling of complex quantities, better declaration capabilities, and optimization features--were added to ALGOL-60. Thus, ALPHA could conveniently be used for solving linear algebraic problems. (36) (37) There is no mention in the literature of any significant problems that were solved by utilizing the ALPHA language. The ALPHA translator for the M-20 allows direct use of matrix notation. ALPHA also uses list structure processing for its internal translating processes, thereby permitting significantly faster compilations; list structure processing, however, is not available in object code. (26) (32) (33)

FORTRAN, developed by IBM, has become the most widely used programming language in the US and much of Europe--in both its original and updated versions. In spite of a late start, the Soviets are now increasing their use of FORTRAN in order to take advantage of the large quantity of commercially available FORTRAN language programs. Since 1967, at least four Soviet FORTRAN compilers have beem developed. One of these is a two-phase compiler written in the Soviet-developed language ALMO (discussed in more detail below). This compiler translates the source language into ALMO and then into machine language. (35)

Both the Joint Institute of Nuclear Research (JINR) at Dubna and the Institute of High Energy Physics at Serpukhov appear to be using FORTRAN as their high level language. (36-38) The FORTRAN compiler at the JINR was developed by converting into BESM-6 code each instruction of a compiler for the US Control Data Corporation 1604 computer. This primitive approach resulted in a much less efficient compiler than one specifically designed to take advantage of the characteristics of the computer with which it is to be used. The Soviets at JINR apparently were willing to sacrifice efficiency in order to utilize immediately the FORTRAN language programs developed for high energy nuclear physics problems at the European Center for Nuclear Research (CERN) in Switzerland. (39) N. N. Gororun of JINR admitted that a major reason for acquiring the CDC 1604 computer was to get the FORTRAN compiler. (40) The JINR also developed the IFVE-67 system for monitoring the performance of FORTRAN language programs, and this system appears to have been incorporated as part of the standard software package for use with the BESM-6 in the solution of engineering as well as nuclear research problems. (41) A major advantage of the FORTRAN language is that it can be used in conjunction with large libraries of relatively independent programs. Other FORTRAN compilers are operational at the Institute of Cybernetics of the Ukrainian Academy of Sciences in Kiev, the Institute of Control Problems in Moscow, and the Novosibirsk State University. (33) The compiler cited in connection with the University was stated to be the Soviet version of FORTRAN-IV and was used with a BESM-6. Only a very limited subroutine library was available as of May 1971; complex variables and double precision variables could not yet be used although procedures for their use were described in the associated manual. (42)

There are numerous examples of Soviet attempts to develop languages for special types of scientific and engineering applications. These developments reflect original theoretical and conceptual work, but in many cases the results do not appear to have been implemented; in certain cases use of the languages is confined to the originating facility. One example is the K-language developed for proving theorems and solving certain classes of linear algebraic equations. This work was based on that of L. V. Kantorovich at the Leningrad Branch of the Mathematics Institute of the Academy of Sciences who pioneered in the use of computers in nonnumerical symbol manipulation. (43) (44) A second example is the SIRIUS language, developed at Kharkov University to perform analytical operations in solving problems of numerical analysis. This was a query-type language that could be used in a conversational mode. A third example is the work of the Institute of Cybernetics in Kiev on the APROKS and SAP-2 languages used for the flame-cutting of ship parts and for chemical plant control, respectively. (45-49) APROKS is currently in use at three Soviet shipyards. (66)

DATA PROCESSING LANGUAGES

Data processing languages were originally designed to solve problems that have large data files on which straightforward operations must be performed. Examples of such problems are payroll, inventory control, and insurance files.

Initially, the USSR exerted little official encouragement to use computers for economic data processing; hence, very little effort on related software was expended until the early 1960s. Even then, after the need for data processing languages was recognized by some Soviet specialists, the amount of effort on data processing languages and compilers remained small. This may be attributed to the fact that Soviet computers and related equipment were not well designed for data processing. In addition, the leading software specialists were largely scientifically oriented. Indeed, Soviet efforts have been aimed at adding data processing features to scientific languages rather than developing languages strictly oriented to data processing. Currently, intensive efforts are under way to develop and use large numbers of computers of appropriate design for economic applications. This will in turn motivate the Soviets toward increased implementation of data processing languages, particularly those already widely used in the West. (51-55)

Only eight of the languages mentioned in the Soviet literature have been categorized as data processing languages. Two of these, FACT and COBOL, originated in the US, and one, A-COBOL, is an extension of COBOL. Until recently COBOL, the most widely used US data processing language, was not widely accepted in the USSR, partially because its terminology was unsatisfactory to many Soviet specialists. Growing Soviet interest in COBOL was indicated in April 1968, when COBOL developers met to formulate an integrated Soviet version of minimal COBOL and to achieve the program compatibility provided in this language. (54) This meeting was evidence of a complete change in Soviet attitude toward the use of COBOL in a growing awareness of the use of the computer as an aid in economic planning and data processing. Three COBOL compilers are presently used for processing large arrays of data in information retrieval applications and economic planning problems. The compiler used for information retrieval was developed at the Institute of Cybernetics, Ukrainian Academy of Sciences, Kiev. (33)

ALGEK is perhaps the most significant and the most publicized data processing language developed by the Soviets. A 1963 directive specified its development for the solution of economic problems. It was produced by computer specialists at Novosibirsk, Moscow, and Kiev. (26) (45) Features were added to ALGOL to accommodate data processing problems and to permit ALGEK to be used in the development of translators. As a result, ALGEK is unduly complex and therefore difficult to use. As late as 1968, it had been used only on a limited scale, probably because of its complex nature. (38) (56-61)

The ALGAMS data processing language, also based on ALGOL, was developed at the Institute of Applied Mathematics. Like FORTRAN-ALMO it too goes through an intermediate ALMO translator to be processed into machine language. (62) The personnel who developed the ALGAMS-ALMO translator are not the same as those who worked on the FORTRAN-ALMO compiler, but these related developments indicate that an exchange of information between the two associated institutes probably took place. (62-65) EPSILON, while classified as a data processing language, has been mentioned in relation to symbol manipulation; it is a machine-oriented language which has been used for software design and implementation as well as for documentation. (12) (34) (50)

STRING AND LIST PROCESSING LANGUAGES

String and list processing languages were originally designed to solve problems that require maintenance of tables of several different kinds of data and/or manipulation of variable length sequences of characters. Compiler writing, theorem proving, manipulation of formal algebraic expressions, linguistic data processing, artificial intelligence, and text processing are all prime examples of potential applications of string and list processing languages. There are five of these languages and three associated Soviet-developed compilers in use in the USSR. Two of the languages (SCL and SIGMA) were developed by the Soviets and the other three (COMIT, SNOBOL, LISP) by the US.

SCL was developed at the Computing Center of the Academy of Sciences in Moscow. It is mentioned in an article by S. S. Lavrov as a string processing language, but insufficient information is available for evaluation. (67)

SIGMA represents the only major attempt by the Soviets to develop a native list processing language; there is no indication of an associated operational compiler. SIGMA was developed at Novosibirsk by A. F. Rar and A. P. Yershov, (68) for use in writing assemblers. (50) The SIGMA language reflects work performed in the US on IPL-V and LISP and concepts borrowed from the US language SLIP.

COMIT was developed in the US, primarily for language translation, but the Soviets have experimented with it. There is no indication of any COMIT compiler in the Soviet Union.

SNOBOL, another US string processing language, utilized experience gained from COMIT. SNOBOL was developed to solve a wide variety of problems involving string handling and pattern matching. The Soviets have indicated a growing interest in SNOBOL and have developed SNOBOL compilers for the BESM-6 computer at the Academy of Sciences Computer Center in Moscow. (33) Other developmental work has been done at the Institute of Computer Design in Moscow. (70)

LISP has been of interest to S. S. Lavrov at the Academy of Sciences Computer Center in Moscow. A compiler was developed there but is reported to be extremely inefficient; it takes 20 minutes to compile fairly small programs. (53) (71) A. P. Yershov [redacted] implemented LISP on a BESM-6 computer, and it has been mentioned also in relation to projects at the Institute of Computer Design in Moscow. (70) (71) (143)

25X1

Although the Soviets are aware of the value of string and list processing languages, they apparently have made no significant advances in this area in spite of publicity given to valuable applications of such languages.

MULTIPURPOSE LANGUAGES

Multipurpose languages were developed so that a single language could meet the needs of several applications. These languages are of special interest because they are the types needed in certain important military applications of computers, including command and control, tactics, logistics, and computer-aided design problems. Some multipurpose languages also can be used reasonably well to develop guidance systems. Fragmentary information indicates that some of the Soviet work on multipurpose languages may have been intended for military applications. (72)

The Soviets have worked on at least five multipurpose languages in addition to the ALGEK language (previously discussed as a data processing language). ALGEK might be considered a multipurpose language in the sense that--like ALGEM--it is based on the kernel of the ALGOL scientific language with added features for data processing.

The ALGEM, ALTEM, ALGOL-COBOL language evolution is of special interest not only because it represents a major continuing Soviet effort but, more importantly, because of some indirect evidence that the leading figures in the effort, A. I. Kitov and F. F. Schiller, have military affiliations. (73) The Soviets have avoided revelation of the full name and location of the institute involved--NIIAA, the Scientific Research Institute of A A--both in reports of the language developments and of a conference on computer-aided design. (74) Kitov has co-authored articles with N. A. Krinitskiy, who has been more directly identified with military uses of computers. In discussions during US visits, Kitov indicated his in-depth knowledge of the use of computers in both Western and Soviet military systems. (75)

25X1 [redacted]

25X1 [redacted] G. A. Mironov, who worked on this military project, was an associate of Krinitskiy.

Compilers for the Minsk-22 computers have been reported for both ALGEM and ALTEM. The latter is an extension of the former and was developed to incorporate elements of the US business-oriented language, COBOL. No compiler for the ALGOL-COBOL extension of the early work has been reported but if, as is suspected, the applications were classified, compilers for such computers as the M-220 model may have been developed and not revealed openly.

Two other centers for work on multipurpose language developments (probably overt) are the Institute of Applied

Mathematics in Moscow and the Computing Center of the Siberian Department of the Academy of Sciences in Novosibirsk. Both groups display a strong interest in PL/1, a US language that has been used in some military tactical and logistics applications. This language is mentioned frequently by A. P. Yershov and a PL/1 manual reportedly was translated into Russian, (72) but there is no indication of an operational PL/1 compiler. S. S. Lavrov, a member of the Working Group on the ALGOL-68 Algorithmic Language, has stated that ALGOL-68 and PL/1 are viewed as possible universal programming languages of a new generation, replacing ALGOL-60, FORTRAN, and COBOL. (78)

In September 1971, the Soviets mentioned that they were planning--and had actually started working on--a single compiler to be used for three different languages: PL/1, SIMULA-67, and ALGOL-68. This was to be a general software package for the BESM-6 that could also be used with the RYAD series of computers (currently being developed). It was further stated that various groups--notably those in Novosibirsk, Moscow, and Leningrad--were cooperating in this project. (79-82) In May 1972 Yershov indicated that this compiler project, being funded through an industrial contract, would probably be completed by 1975. (83) In October he admitted that his work was lagging, apparently because of a lack of programming capabilities. To overcome the problems, an attempt was being made to simplify the program for the language. (121).

[redacted]

25X1

SYSTEM DESIGN LANGUAGES

System design languages are used in the analysis, design, and optimization of systems. Soviet work on such languages has taken two directions. First, there has been work on simulation languages for use in the development of economic planning, industrial control, and for trade, transportation, and communication systems. The languages involved, SIMULA and SIMSCRIPT, are both of Western origin, but only the former appears to have had any extensive uses.

(86) (87) Second, work has been aimed at the development of computers, related components, and software. Eight of the 11 different languages that have been studied for the second type of application were Soviet developments, with varying degrees of dependence on adaptation of prior Western work.

SIMULA is an ALGOL-based simulation language which was developed for a US company by Dr. Kristen Nygaard, Director of the Norwegian Computer Center. Nygaard later provided significant assistance in the implementation and use of SIMULA by Soviet computers. In addition to presenting a number of seminars on the language during visits to the USSR, Nygaard received Soviet specialists at his facility in Norway where he collaborated with them in developing SIMULA compilers for the Ural series of computers during 1967. (88) (89) The Central Economics Mathematics Institute has been a leader in Soviet development of SIMULA language capabilities, probably because no suitable Soviet simulation language was available for use by the Institute in its assignment to develop models of the Soviet economy.

In 1970 the Soviets were reported to have SIMULA compilers for the BESM-3M, URAL-14, and URAL-16 computers and to have a compiler for the BESM-6 under development. (90) Increasing Soviet interest in SIMULA application is exemplified by its incorporation in multipurpose language developments under A. P. Yershov. (79) The Soviets probably prefer SIMULA to other Western-developed simulation languages because of its ready availability and its close relationship to ALGOL, to which they are already committed. In addition to its value in economic applications, SIMULA could be used to develop systems for logistical support and communications.

Although the Soviets have studied and experimented with several Western computer-aided design languages, most of their reported applications have involved languages of their own development. Soviet capabilities for development and use of these languages have been seriously handicapped by the limited storage and input/output capabilities of their domestic computers. Thus, by far the greatest amount of Soviet effort on system design languages has been aimed at development of computers and other electronic components and circuits.

The earliest and the most widely used system design language is the APL-derived language, LYAPAS. (91) (92) This language was developed by A. D. Zakrevskiy and his associates at the Tomsk State University. LYAPAS was specifically developed as a computer-aided design language and reportedly has been implemented on most Soviet computers. It seems to be a highly specialized Boolean logic language. It was kept close to machine language in order to minimize the computer storage required for its use and is therefore very difficult to use. (93-95)

From the standpoint of possible uses in classified projects, probably the most significant Soviet system design language is PROYEKT. It was developed as a cooperative effort of several facilities in the Moscow area under the leadership of N. Ya. Matyukhin and Ye. I. Gurvich; the latter has also experimented with Western-developed languages. As previously noted, people associated with programming developments at the institute designated NIIAA have been assigned to projects which are related to military research and development. Reports from a 1967 seminar chaired by Gurvich revealed an association between computer-aided design work at NIIAA and related work at a facility listed as MNIIP, [redacted] 25X1 to be engaged in military electronic device developments. (MNIIP is believed to be a cover designation for NII-17 in Moscow which has been concerned with computers for use in missile systems. (74) In describing the PROYEKT language in 1967, Matyukhin noted that it was designed to overcome inherent inadequacies which existed in previously designed Soviet languages such as LYAPAS.

PROYEKT was developed on the basis of the general purpose ALGOL-60 language and was designed with adequate computational and string capabilites, good report generation, and strong input/output (I/O) capabilities. The I/O includes alphanumeric printers and plotters. This language was described as suitable for solving problems of synthesizing and modeling digital automata and for solving many algorithms used for structural design. This is perhaps the most advanced language developed by the Soviets for system design efforts.

Results of the PROYEKT language work probably formed the basis for the ASP automated design system described by Matyukhin and Gurvich in 1969 for the M-220 computer. As was the case with the PROYEKT language effort, the ability to produce detailed and thoroughly checked documentation automatically for all phases of the design process has been emphasized in the ASP developments. The ASP system appears to be intended for use in the development of third-generation computers. The Soviet system has incorporated all of the critical features of the Solid Logic Design System (SLDA) developed in 1964 by IBM for use in designing and preparing documentation for the 360-model computers. ASP is called a two-stage system that uses ALGOL-based programming languages. The first stage deals with the functional and structural simulation of the proposed computer and uses a language called MODIS. (96) (97) A second language called MOLK is used for specification and testing of the detailed Boolean logic description of the designs for functional units. The MOLK language also is used for at least part of the second state of the ASP system which specifies the physical implementation of the logical design. The authors claim that the total size of the basic and auxiliary ASP programs is approximately 35,000 instructions. The experiences of IBM with the SLDA system indicate that a much more extensive set of programs for computers with considerably better internal and auxiliary storage capabilities than the M-220 would be needed for the ASP system to be used effectively in the design of any major new computers.

The Institute of Applied Mathematics of the Academy of Sciences in Moscow, which previously has been identified with space and military projects, also has worked on designing languages and compilers apparently intended for classified projects. In 1967, Yu. M. Bayakovskiy described the development of a computer-aided design language called OPAL, which was similar to a Western language specifically intended for the design of parallel processors. In response to questions on the use of the OPAL language, the cryptic comment was made that it had been used by IAM for the design of small parallel computers intended for "unpleasant purposes." (98)

The most publicized Soviet work on software for computer-aided design has been done at the Institute of Cybernetics in Kiev. In addition to having worked with the LYAPAS language, they have also studied and experimented with languages of Western design and origin, notably LOTIS and SLANG. (99) One internally developed language, ALOS, has been used to describe the logic of a computer, and an improved version called ALGORITHM is used in the Institute's PROYEKT computer design system (not to be confused with the PROYEKT language previously discussed). Once again, V. M. Glushkov and his colleagues at the Institute of Cybernetics have chosen to use a language quite different from that of other Soviet groups working on similar problems. Generally, the group in Kiev has concentrated more heavily on a design program rather than on optimization of the language used. In part of their work they have used the assembly language or Autocode for the M-220, and its limitations may have prompted their more recent selection of the higher level ALGORITHM language. Practical results presently within the capabilities of the ALGORITHM language and the PROYEKT system still appear to be at most designs for relatively minor subunits of computers. (100-104)

Other Soviet organizations, such as the Institute of Precision Mechanics and Computing Techniques in Moscow, also have made what appear to be major efforts on computer-aided design, but information is too fragmented for assessment of the software employed. The need to produce good documentation as well as good logical and engineering design for computer equipment is an area commonly emphasized in Soviet work on computer-aided design. The need for using computers to assist in the design, checking, and documentation of software for new computers also is emphasized, but results do not yet appear to be satisfactory. In late 1969, Soviets concerned with programming specialized computers for space acknowledged a lack of software that would allow them to model and debug completely the programs for the specialized computers. (71) Inadequacies in Soviet software for system design also are indicated by failure to develop maintenance programs for the IBM-360 type computers along with the hardware.

MISCELLANEOUS LANGUAGES

The miscellaneous languages include those used for information retrieval, compiler writing, analog computer modeling, and geometry. Six of these languages were developed in the Soviet Union, and a seventh, SYNTOL--an information storage and retrieval language--was developed in France.

The SAPAM language is based on ALGOL-60 (105) and is used for automatic programming of analog machines. A translator has been developed that transforms the mathematical description of a model in SAPAM source language into the language of analog computer components. This could be a powerful tool for use in the support of engineering problems related to military systems development. One indication that SAPAM might be intended for use in classified projects is the apparently deliberate concealment of the affiliation of its developers.

In three separate articles V. F. Turchin briefly described the REFAL language and compiler without supplying any technical details. (106-108) REFAL (recursive functions algorithmic language) was developed at the Institute of Applied Mathematics of the USSR Academy of Sciences in Moscow, apparently for the purpose of writing translators. A REFAL compiler translates the REFAL language into assembly language for the BESM-6 computer.

The ALMO language was developed about 1966 by V. M. Kurorkhin, S. S. Kamynin, E. Z. Lyubimskiy, and V. V. Lutsikovich at the Institute of Applied Mathematics in Moscow. (56)(109)(110) ALMO is a machine-oriented, rather than a problem-oriented, intermediate language and serves as an abstract representation of machines. It is similar in concept to UNCOL (universal computer oriented language) in the United States; however, the US has never successfully developed UNCOL nor any similar language. Apparently each higher level language is to be reduced to ALMO as an intermediate language and then ALMO is to be translated to each machine language. For example, the FORTRAN-ALMO translator translates FORTRAN into ALMO and then to machine language; this compiler is itself written in ALMO. In addition to the FORTRAN-ALMO translator and the previously mentioned ALGAMS-ALMO translator, a COBOL-ALMO translator has been reported to be under development. (111) An ALMO translator has been developed for the BESM-4 with a compilation rate of 120 instructions per minute under poor conditions; a large part of this compiler deals with documentation of the object program.

S. S. Kamynin, E. Z. Lyubimskiy, and A. P. Yershov have been involved in most Soviet efforts to develop an intermediate language. Kamynin and Lyubimskiy were involved in the development of the TA-2 compiler, and Yershov was interested in an intermediate language for ALPHA. (59) (60) (112) (113)

APPLICATIONS POTENTIAL OF LANGUAGES

The strengths and weaknesses of 34* Soviet computer languages are depicted in table 1.

*Some of these languages were excluded because of insufficient data on which to base an evaluation.

The potential usefulness of these languages in certain application areas is depicted in table 2. The figure provides an overview of the Soviet application capabilities and needs reflected in the data presented in table 2. The fact that Soviet computer languages generally have good computational capabilities is reflected in the overwhelming majority of 22 that have been ranked A (have all minimum attributes and excel in at least one) for scientific and technical problem solution. In the areas of command and control, logistics, and tactics, only the Western-developed PL/1 language, which was designed to be machine independent, has been given an A rating; almost 30 have been rated C (do not meet minimum requirements) for use in these areas. The Soviet-developed language PROYEKT and PL/1 have been rated A in the area of computer-aided design; of the remaining 32, 23 have been ranked B (have only the most important attributes) and the remaining 9, C. In the two areas of process control and guidance, the distribution of ratings among the 34 languages is fairly even: 10 As, 15 Bs, and the remaining 9, Cs. It should be noted that PL/1 is the only language which has been rated A in all areas, a good reason why Soviet interest in its use has increased.

Operating Systems

An operating system is composed of that software which controls the computer system's execution of programs. An operating system may provide scheduling, monitoring, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services. Operating systems are designed for a specific model of computer and cannot be easily shared even by users of the same model if minor modifications have been made at the various installations.

When provided, standard operating systems for Soviet serially produced computers generally have been very unsophisticated. No standard software that could properly be called system software is available for the widely used Minsk-22 models although some individual installations have developed primitive types for local needs. Such local modifications probably have slowed the development and use of standard operating systems. Indeed, successful Soviet

TABLE 1

Attributes of Soviet Computer Languages

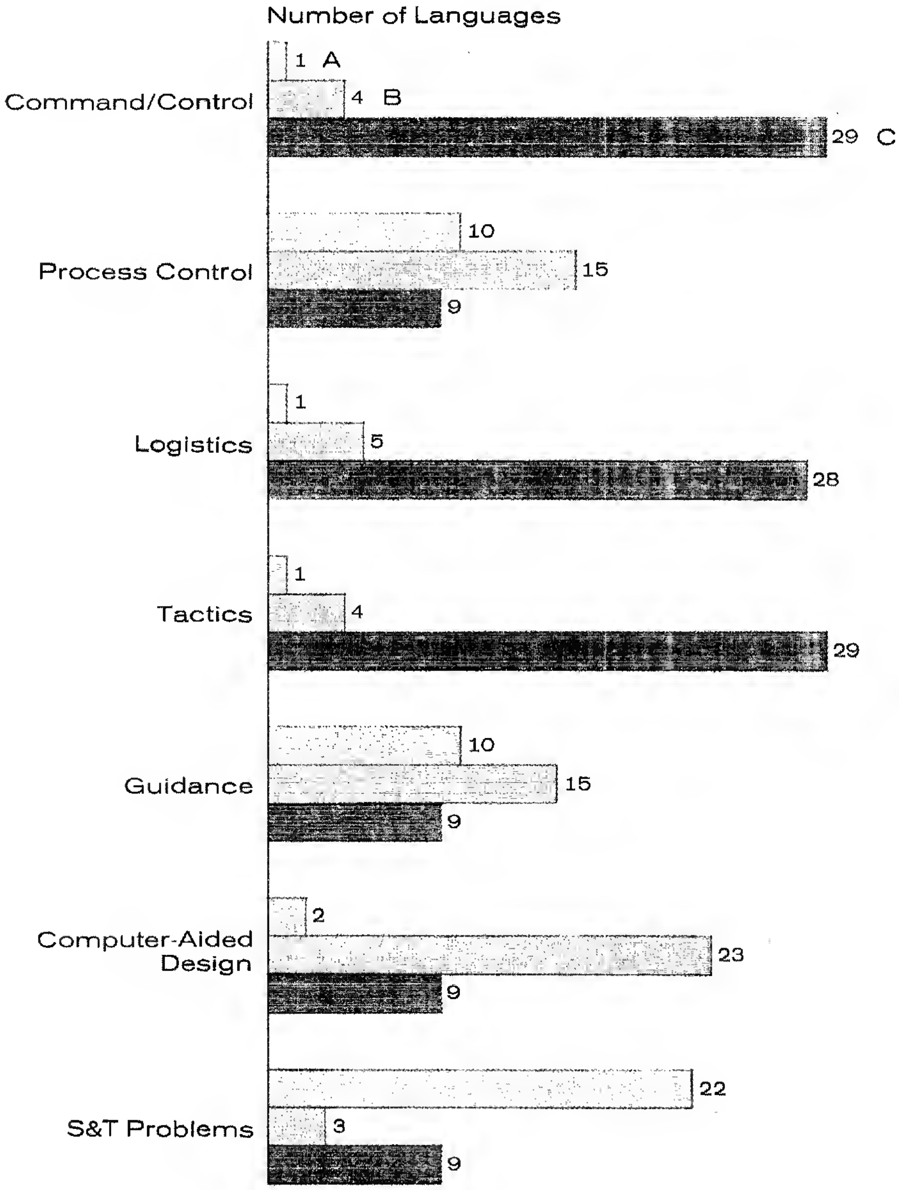
	Attributes				
Language	Input/ Output	String	List	Computation	Report Generation
A-COBOL	good	fair	poor	poor	good
ALCOR	poor	poor	poor	good	poor
ALGAMS	poor	good	poor	good	poor
ALGEK	poor	good	fair	good	poor
ALGEM	poor	good	fair	good	poor
ALGOL	poor	fair	poor	good	fair
ALGOL-COBOL	good	poor	poor	good	good
ALGOS	fair	fair	poor	good	poor
ALMO	fair	good	good	poor	poor
ALPHA	fair	fair	poor	good	poor
ALTEM	poor	good	fair	good	poor
APROKS	fair	poor	poor	good	poor
CERN-FORTRAN	fair	fair	poor	good	poor
COBOL	good	fair	poor	poor	good
COMIT	fair	good	poor	poor	fair
EPSILON	fair	fair	---	good	---
FACT	good	fair	poor	poor	good
FORTRAN	fair	fair	poor	good	poor
K-LANGUAGE	poor	fair	poor	good	---
LISP	poor	fair	good	fair	poor
LOTIS	poor	good	poor	good	---
LYAPAS	poor	good	poor	poor	poor
MALGOL	poor	poor	poor	good	poor
PL/1	good	good	fair	good	good
PROYEKT	fair	good	fair	good	fair
SALGOL	poor	fair	poor	good	poor
SIGMA	poor	fair	good	fair	poor
SIMPOLIZ	poor	poor	poor	good	poor
SIMSCRIPT	fair	fair	fair	good	fair
SIMULA	poor	fair	good	good	poor
SLANG	poor	fair	fair	good	poor
SNOBOL	fair	good	poor	fair	fair
SOL	fair	fair	fair	good	fair
SUBSET-ALGOL	poor	fair	poor	good	poor

TABLE 2

Applications Potential of Soviet Computer Languages

	Application Areas						
Languages	Comm. & Control	Process Control	Logistics	Tactics	Guidance	Comp. Aided Design	S&T Problems
A-COBOL	B	C	B	B	C	C	C
ALCOR	C	B	C	C	B	B	B
ALGAMS	C	B	C	C	B	B	A
ALGEK	C	B	C	C	B	B	A
ALGEM	C	B	C	C	B	B	A
ALGOL	C	B	C	C	B	B	A
ALGOL-COBOL	B	A	B	B	A	B	A
ALGOS	C	A	C	C	A	B	A
ALMO	C	C	C	C	C	C	C
ALPHA	C	A	C	C	A	B	A
ALTEM	C	B	C	C	B	B	A
APROKS	C	B	C	C	B	B	A
CERN-FORTRAN	C	A	C	C	A	B	A
COBOL	B	C	B	B	C	C	C
COMIT	C	C	C	C	C	C	C
EPSILON	C	A	C	C	A	B	A
FACT	B	C	B	B	C	C	C
FORTRAN	C	A	B	C	A	B	A
K-LANGUAGE	C	B	C	C	B	B	A
LISP	C	C	C	C	C	C	C
LOTIS	C	B	C	C	B	B	A
LYAPAS	C	C	C	C	C	C	C
MALGOL	C	B	C	C	B	B	B
PL/1	A	A	A	A	A	A	A
PROYEKT	C	A	C	C	A	A	A
SALGOL	C	B	C	C	B	B	A
SIGMA	C	C	C	C	C	C	C
SIMPOLIZ	C	B	C	C	B	B	B
SIMSCRIPT	C	A	C	C	A	B	A
SIMULA	C	B	C	C	B	B	A
SLANG	C	B	C	C	B	B	A
SNOBOL	C	C	C	C	C	C	C
SOL	C	A	C	C	A	B	A
SUBSET-ALGOL	C	B	C	C	B	B	A

A: has all of the minimum attributes and excels in at least one.
B: has the most important attributes but none of the others.
C: does not meet minimum requirements.



Rated Potential Use of Soviet Computer Languages for Various Applications

514850 11-72 CIA

SECRET

developments in this area are fairly recent; these, too, have been characterized by independent efforts, owing to a lack of uniform input/output devices and limited computer storage capabilities.

BATCHED PROCESSING

The simplest type of computer operating system is one designed to execute one program at a time. Most Soviet computer models--even some of the largest ones--employ this type of operating system, which usually monitors the progress of program performance, permits the operator to interrupt the computation at any stage, and may provide for collection of statistics on job processing. Typical of these is the auto operator system for the M-20 and M-220 models. (114) (115)

Frequently in Soviet discussions on computer applications emphasis has been placed on more efficient techniques for the transfer of blocks of data among internal and auxiliary stores. One simplistic answer to this problem is the D2U system which supervises the transfer of data to and from magnetic tape units of the URAL-11, -14, and -16 computers. (116) An upgraded version of this system, the DIUR, was proposed in 1968, but only the D2U has been noted as standard software with the URAL computers. (117)

In addition to providing for monitoring and record keeping, which are typical features of the batch mode operating system, various local needs have been satisfied by individual developments. Some of these have included the addition of compilers to smaller machines in order to compile programs written in higher level languages for larger machines. For example, the BESM-4 installation at Dubna incorporates an ALGOL compiler to compile programs for the BESM-6. The IFVE-67 monitoring system was developed for a Minsk-22 at the Institute of High Energy Physics at Serpukhov specifically to support a FORTRAN compiler which could compile CERN-FORTRAN programs. (37)

MULTIPROGRAMMING

A multiprogramming system allows concurrent processing of two or more programs by switching between programs during computational delays due to input/output or storage transfers. The BESM-6, which is the largest and fastest Soviet computer to be announced openly, was supposed to accommodate multiprogramming features, but initial efforts were unsuccessful. The Computing Center of the Academy of Sciences in Moscow was responsible for these developments, and its failure may have been due to inadequate manpower assignments and an underestimation of the overall problem.

Although the BESM-6 was available in 1965, the Soviets were still working as late as 1971 to develop a satisfactory operating system. As yet no standard type for all BESM-6 installations has been developed. The criticality of the problem was evident when the job of developing a standard operating system was offered to leading Soviet software specialists, all of whom declined. This task was finally assigned to V. S. Shtarkman at hte Institute of Applied Mathematics, who was diverted from work on the design of computers more advanced than the BESM-6 to do the job. By Shtarkman's own admission, this Institute's operating system for the BESM-6 still is unsatisfactory; it cannot be used with the BESM-6 at the Computing Center of the Academy of Sciences due to differences in the hardware configurations of the BESM-6 installations. (71)

Several other installations have attacked the problem of providing an operating system for the BESM-6 on the basis of their own needs. At the Institute of Cybernetics in Kiev the BESM-6 is multiprogrammed to handle up to three jobs in core; further details are unknown. The BESM-6 operating system at Dubna was developed jointly by the Institute of Precision Mechanics and Computer Techniques, the Joint Institute of Nuclear Research, and the Computer Center of Moscow State University. This operating system uses a virtual memory paging technique for multiprogramming and has an interrupt capability; SUBSET-ALGOL 60 and CERN FORTRAN compilers as well as a SIBFSM-6 assembler are available, and the system relies heavily on subroutine libraries existing in source language, machine language, or SIBESM-6 assembly language. (36) (53) (118-120)

The problems the Soviets have had in developing an operating system for the BESM-6 appear to be due in part to the limited input/output and interrupt capabilities of the machine. Variations among the different installations clearly have made interinstitutional collaboration difficult or impossible.

The Minsk-32 computer is reported to have multiprogramming capabilities. This computer was designed as an upgraded follow-on to the Minsk-22; their programs are compatible. Specifications in 1971 for the Minsk-32 indicated that the central processing unit could accommodate a maximum of four simultaneous jobs residing in its magnetic core memory. Sorting, editing, filing, and assembly/compiling routines reportedly are available, and the general language used is a symbolic one for which individual users have developed a number of macro routines. As with the BESM-6, apparently very little or no compatibility exists among various Minsk-32 operating systems due to tape formatting differences within the various installations (a magnetic tape-based type of operating system is specified for this model of computer). (122)

The Soviets are now attempting to follow the US lead in developing computers patterned after the logical architecture of the IBM/360 series. The first models in the Soviet line, the ASVT (modular computer hardware system) series, were discussed at the first All Union Conference on Programming held in Kiev in 1968. (123) The operating system of the ASVT series is reported to have multiprogramming and multiprocessing capabilities. Economic, process control, and scientific libraries are said to be available, and high level language compilers, an assembler, and debugging and diagnostic utilities are said to exist. (124-126) While the instruction set of the ASVT series appears to be the same as that of the IBM/360 models, the extent of their compatibility is not known, nor is it known whether the ASVT series has the kinds of special instructions that would permit use of IBM/360 compilers and other system software without considerable reprogramming. There are reported to be six models in the ASVT series--M-1000 through M-6000--but only M-4000 through M-6000 are believed to be third-generation computers (i.e., thin- or thick-film hybrid and/or integrated circuits). (127)

TIME-SHARING

A time-sharing system is a multiprogramming system in which multiple users have simultaneous access to the resources of the computer system. This operating system makes it appear to each user that he has a completely dedicated computer system at his disposal.

The Soviets have been trying since the mid-1960s to develop general purpose time-sharing systems for day-to-day use. Indirect evidence indicates that some of their efforts--at least initially--were aimed at military applications, possibly for command and control systems. (128)

The Soviets have claimed that a number of their operating systems have time-sharing capabilities, but the openly reported work on time-sharing systems has been experimental or very simplistic and confined to scientific research and development establishments. One operating system that may incorporate time-sharing features is the SIRENA system for airline reservations. At present, this system reportedly employs some remote terminals with display devices, but Soviet interest in acquiring Western systems for the same purpose indicates that SIRENA is a simplistic first attempt at an on-line time-sharing system. (129) (130) (153)

Of the 24 operating systems identified, eight have been declared by the Soviets to be time-sharing systems. However, the ATSS, UVK, and Dnepr-21 systems seem to be small process control systems with only a crude interrupt capability to simulate a time-sharing environment. (131-134) The TENZOR system, developed in Moscow by M. M. Bezhanova, operates in a dialogue mode to permit several users simultaneously to solve linear algebraic problems, with partially automatic selection of the solution method. (135)

The first Soviet testing of a general purpose time-sharing system was carried out on a Dnepr-2 computer at the Institute of Cybernetics in Kiev around 1969. Teletype-like terminals physically located in the same room as the computer were connected to the computer in a time-sharing mode. It was hoped that the Dnepr-2 would eventually operate as a remote terminal to the larger BESM-6; further specifics or characteristics for this system are lacking. (72) (136)

The operating system developed for the BESM-6 computer by M. R. Shura-Bura and his associates at the Institute of Applied Mathematics in Moscow [redacted] supports a conversational mode via teletypes, but not enough information on this aspect of the system is available for a complete evaluation. (137) (138) 25X1

Perhaps the most widely publicized Soviet attempt to develop a time-sharing system is the AIST (automatic information station system), which was assigned to the Siberian Department of the USSR Academy of Sciences at Novosibirsk. Specifications in 1967 called for this system to service users as far as 200 kilometers from the computer center; it was suspected then that certain users had military affiliations. (128)

The AIST project was broken down into several steps. 25X1 AIST-0 [redacted] supports a conversational ALGOL language (139) and utilizes M-220 computers and a Minsk-22 monitor computer. (111) Software includes system programs, batch processing, console debugging, an incremental compiler, analytic manipulation, and a document search text editor. (31) (140) This software package was estimated to be comprised of 20,000 words. (12) Originally scheduled to be operational in the summer of 1968, (128) AIST-0 was not implemented until June 1970. (139) The next phase in this project is AIST-1, involving two BESM-6 computers with a Ural-14 performing intercommunication functions (141) of collecting and batching data to send to the BESM-6 for standard batch processing. (140) This system is to be located at Akademgorodok and [redacted] will be capable of servicing more than 100 terminals at remote locations. (139) 25X1 Further developments in the AIST project include integration of several BESM-6 computers in order to serve several hundred users. (140) In light of the delays in AIST-0, it is probable that AIST-1 is still in the experimental stage and that more ambitious efforts at time-sharing are only in the design stage.

S/360 TYPE SYSTEMS

As early as 1965 the Soviets began to realize that many advantages could be gained by designing and developing computers similar to those widely used in the West. In that same year they launched the previously mentioned ASVT series, to be used in hierarchical computer systems for data processing in the planning of industrial operations as well as for on-line process control.

The second major effort directed against the S/360 design is the RYAD series of computers being developed jointly by the Soviet Union and the Eastern European countries. A 1968 decision to copy the IBM/360 designs for this project resulted in plans to develop six basic computer models and several modifications (144). While the three larger models probably are still in the design stage--due to the increased number and more advanced design of planned integrated circuits--serial production of the three smaller models and several variations has reportedly begun. There are firm indications that software for the RYAD series of computers probably will be highly oriented toward the PL/1 language.

Developing operating systems for the RYAD series probably will prove to be a troublesome exercise since there are indications that the Soviets have neglected--as they did for the BESM-6--to devote sufficient effort to the problems associated with the Job Control Language, JCL (used to initiate and control job processing and to provide data characteristics and peripheral equipment requirements at the time a program is executed). JCL has been one of the biggest problems for programmers using the complex and advanced MVT S/360 operating system, and the same types of problems would probably exist in using the entire set of S/360 reference manuals. A Soviet translation of the IBM S/360 JCL manual would probably add to the ambiguities already present in the manual. If the Soviets attempt to use a bootlegged "tape" copy of the 360 operating system, they may expect to encounter both operational and technical difficulties--especially if the RYAD hardware is similar but not totally identical to that of S/360, as design specifications indicate.

Modeling computer systems after the S/360 can make available a myriad of related software packages for various applications and thereby reduce design efforts plus the expenditure of time and manpower. However, through naivete and inexperience the Soviets probably will have to solve a great many operating system programming problems before achieving successful operation of the RYAD series of computers.

User Groups

User groups or associations have been conspicuously beneficial to both the developers and employers of computer hardware and software in the West. In both formal meetings and informal communications these groups have shared programs, ideas, and experiences in order to publicize new developments and alert other users to possible problems and corrective measures for their alleviation or circumvention.

Such user group activity in the Soviet Union is probably minimal. In 1968 there was an All-Union Congress on Programming in Kiev, but there was no society in being for programmers; instead, reports were published in _Kibernetika_, a journal of the Academy of Sciences in Kiev. (31) As of 1969 no organization comparable to the US Association for Computing Machinery (ACM) existed in the USSR, although small user groups reportedly were exchanging software. (50) Thus, the numerous gatherings to discuss topics of common concern to computer users appear to be one-time events and not really continuing organizations. (145) (146)

The reason for the small number of formal Soviet user groups probably has been the lack of standardized hardware and software in the Soviet Union. The various hardware configurations of the same model of computer necessitate software variations before a given program can be successfully executed at another location. Also, the independence of computer manufacturers in the Soviet Union allows them to deliver a computer without the responsibility for back-up maintenance support, which in turn forces the end user to develop independently his own software support, a practice which is unheard of in the West.

The only formal Soviet user groups are the associations linked to each of the major computer models produced in the Soviet Union (147-150) plus several miscellaneous groups concerned with computer information dissemination. (148) (151) (152) The largest and best organized group appears to be the Association of Users of the Mir Series of Computers. (150) This association reportedly has a constitution, charges dues, and has 102 organizations as members. Probably the newest group to come into existence is the RYAD Group, formed in 1971 as an intergovernmental association to discuss methods of cooperation in using the RYAD series. (149) This group is probably somewhat premature since the RYAD computer is still being developed and is far from being in day-to-day use. However, its inception and reported existence indicate that the Soviets now recognize that there is much to be gained through standardization, cooperation, and back-up support in the computer hardware and software areas.

TECHNICAL APPENDIX

CHARACTERISTICS OF COMPUTER LANGUAGES

Computer languages can be characterized by certain functions which they support, how well they are capable of performing these functions, and how well suited they are for certain applications. Following are definitions of five attributes, each with three ratings for judging how well individual languages support these attributes. These three ratings are not absolute but relative; they represent a gross way of evaluating a language's capability. These are followed by definitions and general characteristics of seven areas of application and by table 3 which summarizes minimum language attributes desirable for these seven applications.

Language Attributes

Input/Output Capability--The types of input and output devices that can be supported and the convenient use of the devices.

1) Poor--The language supports only a very basic I/O capability, such as paper tape input and output or specialized sensor devices.

2) Fair--The language support includes the basic capability plus magnetic tape handling and a printer capability.

3) Good--The language supports basic I/O, tape handling, printer capabilities and random access devices.

String Processing--The capability to handle alphanumeric character representations in nonnumeric operations.

1) Poor--The language can support only fixed length strings in a few basic operations.

2) Fair--The language can support fixed and variable length strings in a few basic operations.

3) Good--The language can support fixed and variable length strings in a wide assortment of operations.

List Processing--The ability to list or chain together logically related alphanumeric character strings.

1) Poor--The language supports no list capabilities explicitly but does allow programmers to create their own form of list processing.

2) Fair--The language supports basic list processing with a few basic operations.

3) Good--The language supports list processing very easily and contains many operations.

Computations--The ability to perform mathematical operations.

1) Poor--The language supports basic operations (addition, subtraction, multiplication, division) in perhaps one number form (i.e., floating point only).

2) Fair--The language supports the basic operations in both integer and floating point representation.

3) Good--The language supports basic operations in multiple representations and easily describes mathematical calculations. A large mathematical subroutine library is included.

Report_Generation--The ability to write "management type" reports including both computational results and alphanumeric information.

1) Poor--The language supports a very basic capability for printing in which a programmer has to worry about many details.

2) Fair--The language supports a basic capability and includes aids to simplify the programmer's effort.

3) Good--The language supports a sophisticated capability making it very easy to write reports.

Application_Areas_and_Attributes

Command_and_Control--A computer system which supports various levels of military personnel in a chain of command by providing information necessary to carry out their objectives. A command and control system relies on such operational subsystems as logistics and tactical systems for information. It is a military management information system with the following characteristics:
1) Large amount of data,
2) Relatively simple calculations,
3) Complex data handling requirements, and
4) Complex report generation.

Tactical_System--A computer system which aids in the arranging, positioning, or maneuvering of forces in contact, or near-contact, with the enemy so as to achieve the objectives in an air or surface battle. Characteristics are:

1) Storage of vast amounts of data,
2) Retrieval of vast amounts of data in real time,
3) Formatting and printing of small amounts of data, and
4) Deleting and adding elements to large lists of data in real time.

Logistics System--A computer system that facilitates the support given by a command (or other) organization to a person, activity, unit, or force in which all or any part of its supplies, equipment, combat materiel, maintenance, transportation, administration, or any other service is furnished to enable the person, activity, unit, or force to carry out its own operation more expeditiously. Characteristics are:

1) Storage of vast amounts of data,
2) Retrieval of vast amounts of data in a reasonable time period,
3) Complexity of interrelated items,
4) Formatting and printing of large amounts of data, and
5) Deleting and adding elements to large lists of data in a reasonable time period, i.e., one to seven days.

Guidance System--A computer system which completely or partially controls the act of guiding a moving object (e.g., rockets, bombs, or other missiles) along a course. Characteristics are:

1) Real time,
2) Storage of a small volume of data, and
3) Complicated mathematical calculations.

Process Control System--A computer system used in controlling some technological process (e.g., petroleum refining or steel making). An advanced process control system is a closed system in which inputs are evaluated and an optimal output command (e.g., change the temperature to 115 degrees) is sent electronically without human intervention. An elementary type is an open system in which the suggested output command is printed or displayed and human intervention is required actually to execute the command, (e.g., turn the burners up to increase heat). Characteristics are:

1) Real time,
2) Storage of a small volume of data,
3) Simple comparisons of values against object values, and
4) Complex calculations (required for the advanced type).

Computer-Aided Design System (digital computers)--A computer system used to facilitate the design and implementation of digital computers. Characteristics are:

1) Storage of large amounts of data,
2) Complicated mathematical and logical computations, and
3) Good report generation capabilities.

<u>Scientific and Technological Problems</u>--Problems concerned with complex computations such as matrix inversion, evaluation of mathematical formulas, and engineering calculations. Characteristics are:

1) Minimal input/output,
2) Vast amounts of computation, and
3) Processing of a small volume of data.

Table 3 summarizes the minimum language attributes desirable for these application areas.

TABLE 3

Minimum Language Attributes Desirable for Certain Applications

Application Areas	Input/ Output	String	List	Computation	Report Generation
Command & control	good*	fair	fair	fair	good*
Process control	fair	poor	poor	good*	poor
Logistics	good*	fair	fair	fair	good*
Tactics	good*	fair	fair	fair	fair
Guidance	fair	poor	poor	good*	poor
Computer-aided design	fair	fair	fair	good*	fair
Science & Technology	poor	poor	poor	good*	poor

*Indicates that the attribute is very important for the application area.

UNCLASSIFIED REFERENCES

The following abbreviations have been used for publications cited in these references:

Abbreviation	Russian Title	English Translation
DAN	Doklady Akademii Nauk SSSR	Report of the Academy of Sciences, USSR
K	Kibernetika	Cybernetics
TPVKpoP	Turdy Pervoy Vsesoyuznoy Konferentsii po Programmirovaniyu	Proceedings of the First All-Union Conference on Programming
TVVKpoP	Trudy Vtoroy Vsesoyuznoy Konferentsii po Programmirovaniyu	Proceedings of the Second All-Union Conference on Programming
VAN	Vestnik Akademii Nauk SSSR	Herald of the Academy of Sciences, USSR
ZhVMiMF	Zhurnal Vychislitel'noy Matematiki i Matematicheskoy Fiziki	Journal of Computer Mathematics and Mathematical Physics

1. Yershov, A. P. "Fundamental Principles for the Setup of a Programming Program at the Mathematical Institute of the Siberian Department of the Academy of Sciences, USSR," Sibirskiy Matematicheskiy Zhurnal, v 2, no 6, 1961, p 835-852

2. Yershov, A. P. "Operator Algorithms (Basic Concepts) Part I," Problemy Kibernetiki, no 3, 1960, p 5-48

3. D'yachenko, V. F. "Transformation of Logical Systems of Algorithms," Problemy Peredachi Informatsii, no 17, p 35-47

4. Yanov, Yu. I. "On the Equality of Program Schemes and Their Transformations," DAN, v 113, no 1, 1957, p 39-42

5. Bekishev, G. A. "Deparallelization of Computer Algorithms," Vychislitel'nyye Sistemy, no 5, 1963, p 22-30

6. Yershov, A. P. "Reducing the Problem of Memory Allocation During Program Compiling to the Problem of Coloring Chart Vertices," DAN, v 142, no 4, 1 Feb 62, p 785-789

7. Neyman, V. I., and Paramonov, Yu. V. "On a Method for

Obtaining Random Numbers," Problemy Peredachi Informatsii, no 11, 1962

8. Lyapunov, A. A. "An Algebraic Treatment of Programming," Problemy Kibernetiki, no 8, 1962, p 235-241

9. Kitov, Anatoliy Ivanovich, and Krinitskiy, Nikolay Andreyevich Electronic Digital Computers and Programming, 2nd ed., Moscow, Fizmatgiz, 1961

11. Yershov, A. P., Kozhukhin, G. I., and Voloshin, Yu. M. "Input Language for Automatic Programming Systems," Referativnyy Zhurnal-Kibernetika, no 7, Jul 64

13. Popov, V. N., Stepanov, V. A., Stisheva, G. A., and Travnikova, N. A. "A Compiler," ZhVMiMF, v 4, no 1, Jan-Feb 64, p 78-95

14. Gladun, V. P., Letichevskiy, A. A., Mikhnovskiy, S. D., and Rabinovich, Z. L. "Expanded ALGOL," ZhVMiMF, v 5, no 2, Mar-Apr 65, p 369-372

15. Shura-Bura, M. R., and Lyubimskiy, E. Z. "An ALGOL-60 Converter," ZhVMiMF, v 4, no 1, Jan-Feb 64, p 96-112

16. Shura-Bura, M. R. "International Scientific Life," VAN, no 11, Nov 60, p 94-97

17. Glushkov, V. M. "On a Signle Synthesis Algorithm for Abstract Automats," Ukrainskiy Matematicheskiy Zhurnal, v 12, no 2, Apr 60, p 147-156

18. Yershov, A. P., Lavrov, S. S., and Shura-Bura, M. R. "Input-Output Procedures in the ALGOL-60 Language," ZhVMiMF, v 4, no 5, p 970-973

19. Shura-Bura, M. R. "Programming," Mathematics in USSR Over Forty Years, 1917-1957, Moscow, v 1, 1959, p 879-886

20. "Development of Research on the Automation of Programming," VAN no 5, 1964, p 19-26

21. Smirnova, T. N. "A Polynomial Compiler and the Performance of Analytic Transformations with the Aid of an Electronic Computer," Trudy Matematicheskoy Instituta, Moscow, v 66, 1962

22. Morozov, Yu. I. "Programming System of the Compiling Type (PSK-1)," Referativnyy Zhurnal--Matematika, no 6, 1962

23. Arin', E. I. "The Symbolic Address Method for a Two-Address Computer," Referativnyy Zhurnal--Avtomatika i Radioelektronika, no 12, 1961

24. Krinitskiy, N. A., Mironov, G. A., and Fiolov, G. D. Programming, Moscow, Fizmatgiz, 1963

35. Bukhshtab, Yu. A., Ermakov, E. A., Koryagin, D. A.,

Levina, A. M., and Maklahov, A. V. "FORTRAN-ALMO System of Programming," TPVKpoP, 1968

36. Govorun, N. N., Veretenov, V. Yu., Volkov, A. I., Zaikin, N. S., Silin, I. N., Fedorova, R. N., and Shirikov, V. P. "The Dubna Monitoring System for the BESM-6," TVVKpoP, Feb 70, p 3-6

37. Sokolov, S. N., Kalinchenko, P. A., Lupashina, I. S., Makarov, V. A., Markov, A. S., and Popova, I. V. "The FORTRAN IFVE Translator," TPVKpoP, 1968

38. Shurakov, V. V., and Morozov, V. P. "An Automatic Programming System for Data Processing Problems on the Minsk-22 Computer," TVVKpoP, Feb 70, p 3-6

41. Sokolov, S. N., Kalinchenko, P. A., Lupashina, I. S., Makarov, V. A., Markov, A. S., and Popova, I. V. "Monitoring System IFVE-67," TPVKpoP, 1968

43. Smirnova, T. N., and Shakhbazyan, K. V. "Project of K-Translator," TPVKpoP, 1968

44. Shakhbazyan, K. V. "The K-Language and Parallel Branches of Algorithms, TPVKpoP, 1968

45. Tamm, B. G., Pruuden, Yu. I., and Pruuden, E. V. Automation of the Preparation of Programs for Metal Cutting Machines with the Help of Computers, 1966

46. Tamm, B. G. "Problems of Programming Automation in the Estonian SSR." Izvestiya Akademii Nauk Estonskoy SSR--Fizikomatematika, v 16, no 3, 1967, p 267-284

47. Tamm, B. G. "Problem-Oriented Programming Systems in Engineering," Izvestiya Akademii Nauk Estonskoy SSR--Fizikomatematika, v 17, no 3, 1968, p 260-269

48. Aksel'rod, I. R , and Belons, L. F. "SIRIUS-Input Language for an Automatic Programming System," Doklady Seminariya po Avtomatizatsiyu Programmirovanii, no 3, 1967, p 44-62

51. Romanovskaya, L. M., Stolyarov, G. K., Fedorov, A. T., and Fel'dman, L. S. "An Automatic Programming System Based on a Compatible Minimal COBOL Language," TVVKpoP, Feb 70, p 3-6

52. Volkova, N. A. "Organization of Data Input/Output in the COBOL-DNEPR-2 Translator," Doklady Seminariya po Avtomatizatsiyu Programmirovanii, no 3, 1967, p 16-21

53. Boehm, B. W. "Extensive Tour Yields Report on Current Soviet Computing," Soviet Cybernetics Review, v 1, no 1, Jan 71

54. Babenko, L. P., Romanovskaya, L. M., Stolyarov, G. K., and Yushchenko, E. L. "A Compatible Minimal COBOL for

Domestic Serial Computers," TPVKpoP, 1968

55. Malinovkiy, B. N., et al. "The DNEPR-2 System," Fundamentals of Designing Industrial Control Machines, Mashinostroyeniye, Moscow, ch 1, sec 2, 1969, p 28-36

56. Red'ko, V. N., et al. K, no 5, 1967, p 87-91

57. Rand Corp. RM-5136-PR, Soviet Cybernetics, Dec 66

58. Rand Corp. RM-5157-PR, Soviet Cybernetics, Feb 67

59. Yershov, A. P. "Algorithmic Programming Languages," VAN, no 3, Mar 68, p 58

60. Yershov, A. P. "An Automatic Programming System of High Efficiency," Journal of the Association for Computer Machinery, v 13, no 1, Jan 66, p 17-24

61. Dorodnitsyna, A. A., Gerlakh, and Chaikouskiy, L. F. Abstract no 06539-67, 21 Mar 67

62. Lutsikovich, V. V., Pastukhova, L. I., Nemirovskaya, L. M., Ukhov, L. V., and Fol'teni, V. "The ALGAMS-ALMO Translator," TPVKpoP, 1968

63. "Description of the ALGAMS Language," Algoritmy i Algoritmicheskiye Yaziki, no 3, 1967, p 3-13

64. Peledov, G. V., Kalitin, S. S., Besfamil'nyy, M. S., Maksimenkova, A. S., Kartseva, M. V., Churkina, L. V., and Fomin, L. M. "The MEI-2 ALGAMS Syntactical Translator for the Minsk-22 Computer," TPVKpoP, 1968

65. Shkut, N. V. "Problems of Optimizing Operating Programs in the TAM-22T and TAM-32 Translators," TVVKpoP, Feb 1970, p 3-6

67. Lavrov, S. S. "SNOBOL Adopted for Soviet Computers," SNOBOL-A--A String Manipulation Language, Computer Center, USSR Academy of Sciences, Moscow, 1968, p 3-15

68. Yershov, A. P., and Rar, A. F. "SIGMA Computer Language Developed at Novosibirsk," FTD Bulletin, FTD-CP-17-1-67, 28 Apr 67

73. Kitov, A. I., Mazeyev, M. Ya., and Shiller, F. F. "The ALGOL-COBOL Algorithmic Language," TPVKpoP, 1968

74. K, no 3, 1968, p 102

77. Shiller, F. F., and Kiryukhin, Yu. P. "An Automatic Programming System for Economic and Mathematical Problems on the Minsk-22 Computer," TPVKpoP, 1968

78. Hoare, C. A. R., Lavrov, S. S., and Kurochkin, V. M. "Working Session IV: The Use of High Level Languages in Large Program Construction," Efficient Production of Large

Programs, from the Proceedings of the International Workshop at Jablonna, 10-14 Aug 70

86. Daugavet, O. K., and Klokachev, I. V. "Translators of Algorithms from the SIMULA Modeling Language," TPVKpoP, 1968

87. Daugavet, O. K., Igolkina, D. V., Klokachev, I. V., and Sharaya, E. Yu. "Translation of the SIMULA Simulation Language into ALGOL-60," TVVKpoP, Feb 70, p 3-6

89. DOD IIR 1 872 0128 67, 26 Jul 67

91. Gavrilov, M. A., and Zakrevskiy, A. D. A Programming Language for Logic and Coding Algorithms, Academic Press, New York and London, 1969

92. Utkin, A. A. "Simulation of Discrete Automata on General-Purpose Computers," Referativnyy Zhurnal--Kibernetika, Abs. 7G173

94. Nadler, Morton. Computer, Jul-Aug 71, p 36

96. Matyukhin, N. Ya., Engalychev, A. M., Gurvich, E. I., Afanas'yeva, N. Z., Krapchin, A. I., and Oleynik, R. I. "Modeling Digital Devices," The Application of the Computer to the Design of Digital Devices, Moscow, 1968, p 47

97. Gurvich, E. I., and Matyukhin, N. Ya. "The ASP Automated Computer Design System," Automation and Computer Technology, no 5, 1969, p 1-6

99. Kalinichenko, L. A., and Moskalenko, V. M. "Study of a Time-Sharing System by Means of a Simulation Model," TVVKpoP, Feb 70, p 3-6

101. Glushkov, V. M., Kapitonova, Yu. V., and Letichevskiy, A. A. "Glushkov Describes the Proyekt Design System," Soviet Cybernetics Review, Jan 72, p 23-28

102. Glushkov, V. M., Kapitonov, Yu. V., and Letichevskiy, A. A. "The Types of Language Used for Describing Data in the Automated System for Planning Computers," K, no 6, 1970, p 3ff

103. Glushkov, V. M., Kapitonov, Yu. V., and Letichevskiy, A. A "The Method of Planning Computers in the Proyekt System," K, no 2, 1971, p 1ff

104. Letichevskiy, A. A., Grishchenko, N. M., Fedyurko, V. V., Felizhanko, O. D., and Sherctoboyev, G. G. "The Basic System for Automating Programming in the Proyekt System," K, no 6, 1971, p 72-74

105. Bereslavskiy, A. Kh., and Gol'denberg, G. S. "SAPAM--A System for Automatic Programming of Analog Machines," K, no 2, 1968, p 36-44

106. Turchin, V. F. "Translator of ALGOL Written in the

REFAL Language," TPVKpoP, 1968

107. Turchin, V. F. "An ALGOL Translator Written in the REFAL Language," The REFAL Recursive Functions Algorithmic Language, Sep 68, p 3-6

108. Florentsev, S. N., Olyunin, V. Yu., and Turchin, V. F. "A REFAL Interpreter," TPVKpoP, 1968

109. Kulik, V. T., and Saprykin, V. I. "Use of the ALOS Language for Describing Arithmetic Algorithms," tr Kiyevsk. Politekhn. in-ta, v 42, 1963, p 104-113

110. Barkova, I. S., Bogdanov, V. V., Isayenko, T. G., Kamynin, S. S., Konovalov, N. A., Levinson, E. D., Lyubimskiy, E. Z., Semyachkin, V. A., Ushkova, V. L., and Shtarkman, V. S. "An ALMO Compiler for the BESM-4 Computer," TPVKpoP, 1968

112. Kamynin, S. S., and Lyubimskiy, E. Z. "Procedure Codes in the TA-2 Translator," ZhVMiMF, v 5, no 4, 1965, p 699-708

113. Yershov, A. P., and Yushchenko, E. L. "New Report on the First All-Union Conference on Programming," K, no 3, May-Jun 69, p 101-102

114. Surzhikov, S. P. "Automation of a Machine of a Middle Class of the M-20 Type (Auto Operators 2)," TPVKpoP, 1968

115. Maksimey, I. V., and Surzhikov, S. P. "Organizing the Collection of Statistics on the Structure of Jobs Using the Auto-Operator for the M-20 Computer," TPVKpoP, 1968

116. "Supervisory Program Language D2U (Ural General-Purpose Digital Computers)," unpublished MS, Ministry of USSR Radio Industry, PSO. 170.102, I68 Ed., 1968

117. Geronimus, Yu. V. "Design Plans for the Dispatcher DIUR-14-2 for the Computer Ural-14," TPVKpoP, 1968

118. Veretenov, V. Yu., Govorun, N. N., Zhogolev, E. A., Ivannikov, V. P., Kabanov, M. I., Korolev, L. N., Pahskin, E. N., Podshivalov, D. B., Rau, O. I., Rostovtsev, V. A., Silin, I. N., Tomilin, A. N., Trifonov, N. P., Chaykovskiy, M. G., and Shirikov, V. P. "A Variation of the Operating System for a Production Model of the BEMS-6 Computer," TPVKpoP, 1968

119. Zhogolev, E. A. "Automatic Programming System for the BESM-6 Computer," Computational Techniques and Programming, University Publishing House, Moscow, v 9, 1967, p 2-17

120. Korolev, L. N., Ivannikov, V. P., and Tomilin, A. N. "Functions of the Dispatcher of the Operating System of the BESM-6," ZhVMiMF, v 8, no 6, 1968, p 1403-1418

123. Khayet, V. S., and Shcherbakov, Ye. V. "A Functional Structure of the Operational System ASVT," TPVKpoP, 1968

124. Gegeshidze, Professor G. A. "The Development of New Means of Computer Technology," Instruments and Control Systems, no 4, 1970, p 16-18

125. Kostelyanskiy, V M., and Barabanov, V. A. "ASVT Modular Computer System," Computer Hardware for Production Control, Mashinostroyeniye, Moscow, ch 3, 1969, p 177-182

126. Vorozheykin, V. V. "Editing Program for Computer Software for the ASVT (RP-ASVT)," TPVKpoP, 1968

127. Presse der Sovjet Union Industrie, no 91, 1969

129. Aviation Week and Space Technology, 31 Jul 72, p 14

130. Interavia News Letter, no 7535, 21 Jun 72

131. Voskresenskiy, Yu. A., Kozmidiadi, V. A., Kozmidiadi, N. P., Olefir, L. V., and Smakotina, T. A. "The ATSS Special-Purpose Organizing System," TVVKopP, Feb 70, p 3-6.

132. Berezin, V. N. and Gushchin, S. N. "Program for Coordinating and Controlling the Computation and Exchange in Real Time for a Controlling Computer Complex," TPVKpoP, 1968

133. Trakhtengerts, E. A., and Yurchenko, V. Ye. The Order Code for Computer Control, Izd-vo Nauka, Moscow, 1965, p 101-112

134. Keldysh, M. V. The opening address of the General Meeting of the USSR Academy of Sciences, 6-7 Mar 68, VAN, no 5, 1968, p 3-43

135. Bezhanova, M. M. "The Tenzor System Program," TVVKpoP, Feb 70, p 3-6

137. Zadykhaylo, I. B., Kamynin, S. S., Lyubimskiy, E. A., and Shura-Bura, M. R. "The BESM-6 Operating System of the USSR Academy of Sciences," Institute of Applied Mathematics, TVVKpoP, Feb 70, p 3-6

138. Bakharev, I. A., Lyubimskiy, E. Z., and Tyurin, V. F. "Organization of Teletype Operations and Debugging in the IAM Operating System," TVVKpoP, Feb 70, p 3-6

142. Korolev, M. A. "Report on the Algorithmic Language for Economic Calculations, ALGEK," Moscow Economic-Statistics Institute, 19 Jun 64

145. "Ukrainian Republic Collection of Algorithms and Programs," Institute of Cybernetics, AS Ukr SSR, Kiev, 1969

146. D'yachenko, A. I. "Ukrainian Republic Fund of Algorithms and Programs," Mekhanizatsiya i Avtomatizatsiya

Kontrola, no 1, 1970, p 61

147. "Minsk and Ural Computer Users Groups," Nauchnotekhnicheskaya Informatsiya, series 2, no 2, 1967, p 45

148. Holland, Wade B. "The BESM-6 Computer," Datamation, Aug 67

149. "In Socialist Countries," Sotsialisticheskaya Industriya, 18 Apr 71, p 3

150. "Mir Users Group Established," Vestnik Akademii Nauk Ukrainskoy SSR, no 5, May 70, p 38

151. "The Electronic Computer Center of Estonia," Sotsialisticheskaya Industriya, 10 Jan 71, p 2

152. Zlochevskiy et al. "State-Wide Information System Sets Goals," Information in Scientific Studies, Naukova Dumka, Kiev, 1969, p 224-229

153. Zhozhikashvili, V. A., Mitzkevich, L. V., Novokhatniyy, A. A., and Silayav, V. N. "Teleavtomaticheskaya Systema Massovogo Obsluzhivaniya 'SIRENA'," Instruments and Systems of Control, no 9, 1970, p 1-4

25X1

25X1

[redacted]

Secret

25X1

Next 4 Page(s) In Document Exempt